TM291 Management Information System

MEETING 2: DATA DESIGN CONCEPTS

AOU UNIVERSITY

Data Design Concepts

Before constructing an information system, a systems analyst must understand basic design concepts, including data structures and the characteristics of file processing and database systems, including Web-based database design

Data Design Concepts

The data storage function manages how data is stored and handled by programs that run the system

Goals of data storage design

- Efficient data retrieval (good response time)
- Access to the information users need

Data Structures

- A framework for organizing and storing data
- Data Structures consist of files or tables that are linked in various ways
  - A file or table contains data about people, places or events or any important data that the business must kept information about
  - Depending on how the system's file and tables are organized and linked, an (IS) is called either
  - File-oriented system: also called a file processing system, stores and manages data in one or more separate files.
  - Database system: consists of linked data files, also called tables, that form an overall data structure. Compared to file processing, a database environment offers greater flexibility and efficiency.

Overview of File Processing

- Companies mainly use file processing to handle large volumes of structured data on a regular basis
- File processing design approach was well suited to mainframe hardware and batch input
- File processing design is less common today, file processing can be more efficient and cost less than a DBMS in certain situations

Overview of File Processing

File Processing Advantages

- Simplicity: the design of file processing is more simple than designing Database
- Efficiency: file processing cost less and can be more speed than Database
- Customization: you can customize file processing more easily and efficiently than Database because files are related with the application and it have all the data needed for that application

Overview of File Processing

- Potential problems
  - Data redundancy: occurs when data common to two or more information systems is stored in several places. Data redundancy requires more storage space, and maintaining and updating data in several locations is expensive.
  - Data integrity: Refers to the validity of data. Data integrity can be compromised in a number of ways: human errors when data is entered, errors that occur when data is transmitted from one computer to another, software bugs or viruses, hardware malfunctions, such as disk crashes and natural disasters, such as fires and floods.
  - Rigid data structure: A data structure that is hard to work with and inflexible. File-processing is rigid when compared to a typical database management system.

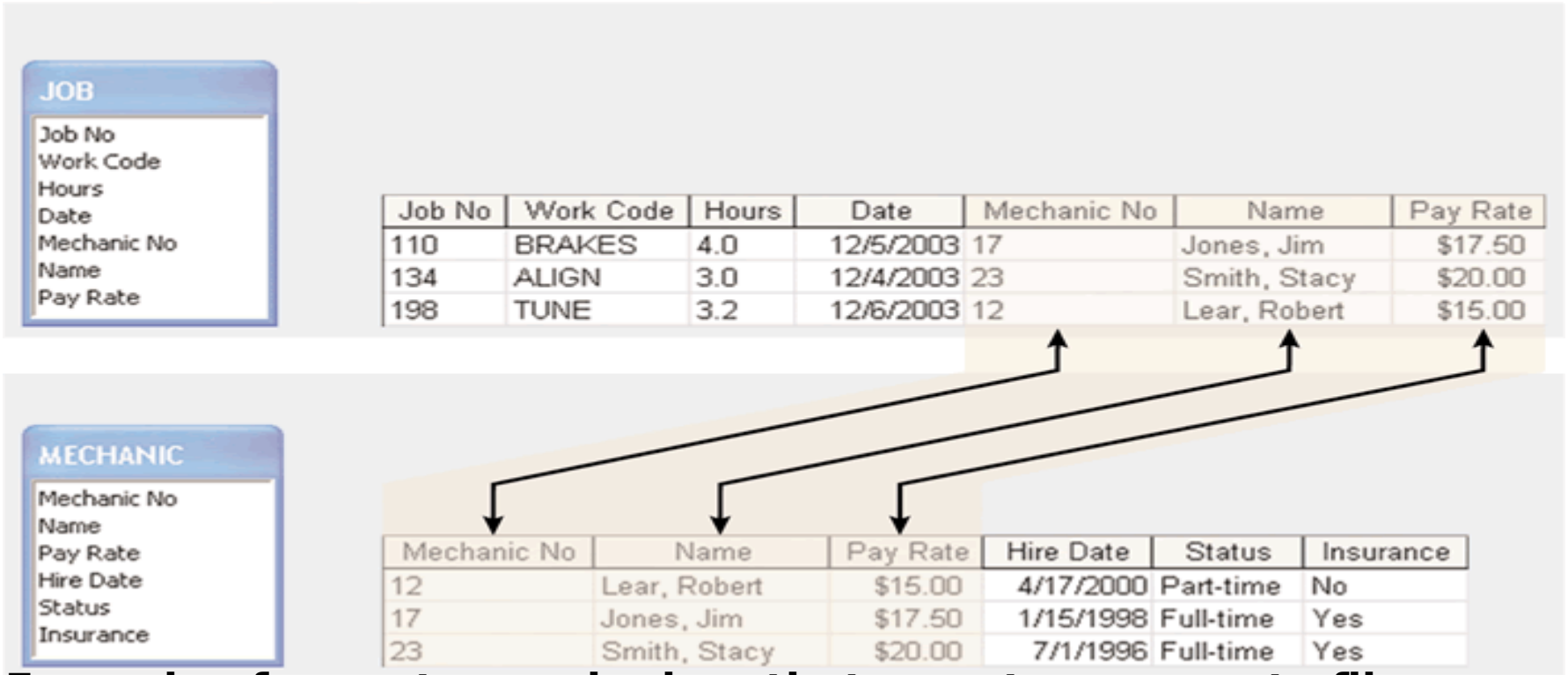
Overview of File Processing

- Uses various types of files
  - Master file: In a typical file processing environment, a master file stores relatively permanent data about an entity. For example, a PRODUCT master file might contain one logical record for each product a company sells.
  - Table file: contains reference data used by the (IS). As with master files, table files are relatively static and are not updated by the (IS). Examples include tax tables and postage rate tables
  - Transaction file: In a typical file processing environment, a transaction file stores records that contain day-to-day business and operational data. A transaction file is an input file that updates a master file; after the update is completed, the transaction file has served its purpose.

Overview of File Processing

- Uses various types of files
  - Work file – scratch file: In a typical file processing environment, a work file is a temporary file created by an information system for a single task. Most often a work file is created by one process in the information system and used by another process within the same system. Work files also are called scratch files.
  - Security file: Sequence codes are numbers or letters assigned in a specific order. Sequence codes contain no additional information other than an indication of order of entry into a system.
  - History file: In a typical file processing environment, a history file is a file copy created and saved for historical or archiving purposes. New history files, unlike new security files, do not replace the old files.

Example of File Processing of an auto repair shop



Job No	Work Code	Hours	Date	Mechanic No	Name	Pay Rate
110	BRAKES	4.0	12/5/2003	17	Jones, Jim	$17.50
134	ALIGN	3.0	12/4/2003	23	Smith, Stacy	$20.00
198	TUNE	3.2	12/6/2003	12	Lear, Robert	$15.00

Mechanic No	Name	Pay Rate	Hire Date	Status	Insurance
12	Lear, Robert	$15.00	4/17/2000	Part-time	No
17	Jones, Jim	$17.50	1/15/1998	Full-time	Yes
23	Smith, Stacy	$20.00	7/1/1996	Full-time	Yes

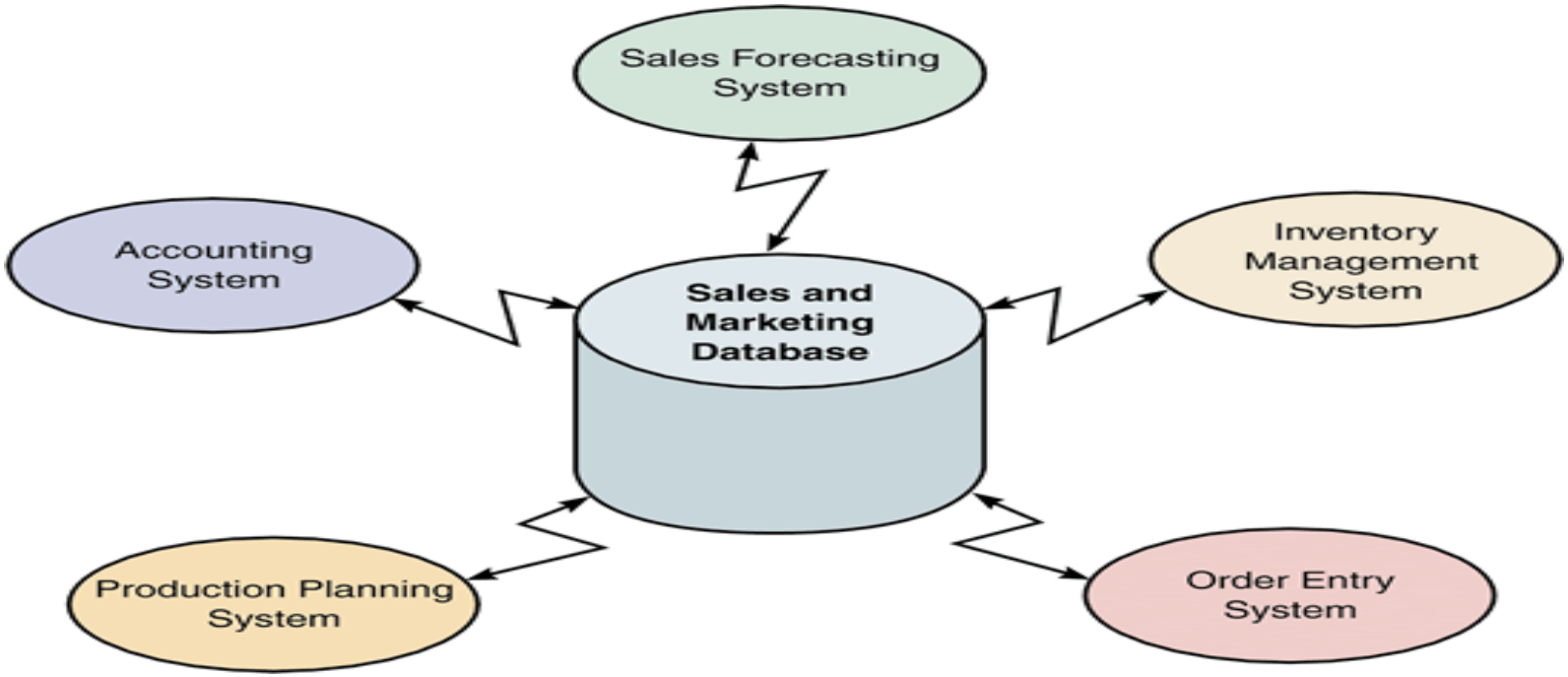
Example of an auto repair shop that uses two separate file-oriented systems: a Job Records System (with a JOB data file) and an Employee Records System (with a MECHANIC data file). Notice that three items of information must be duplicated in both data files.

Overview of Database Systems

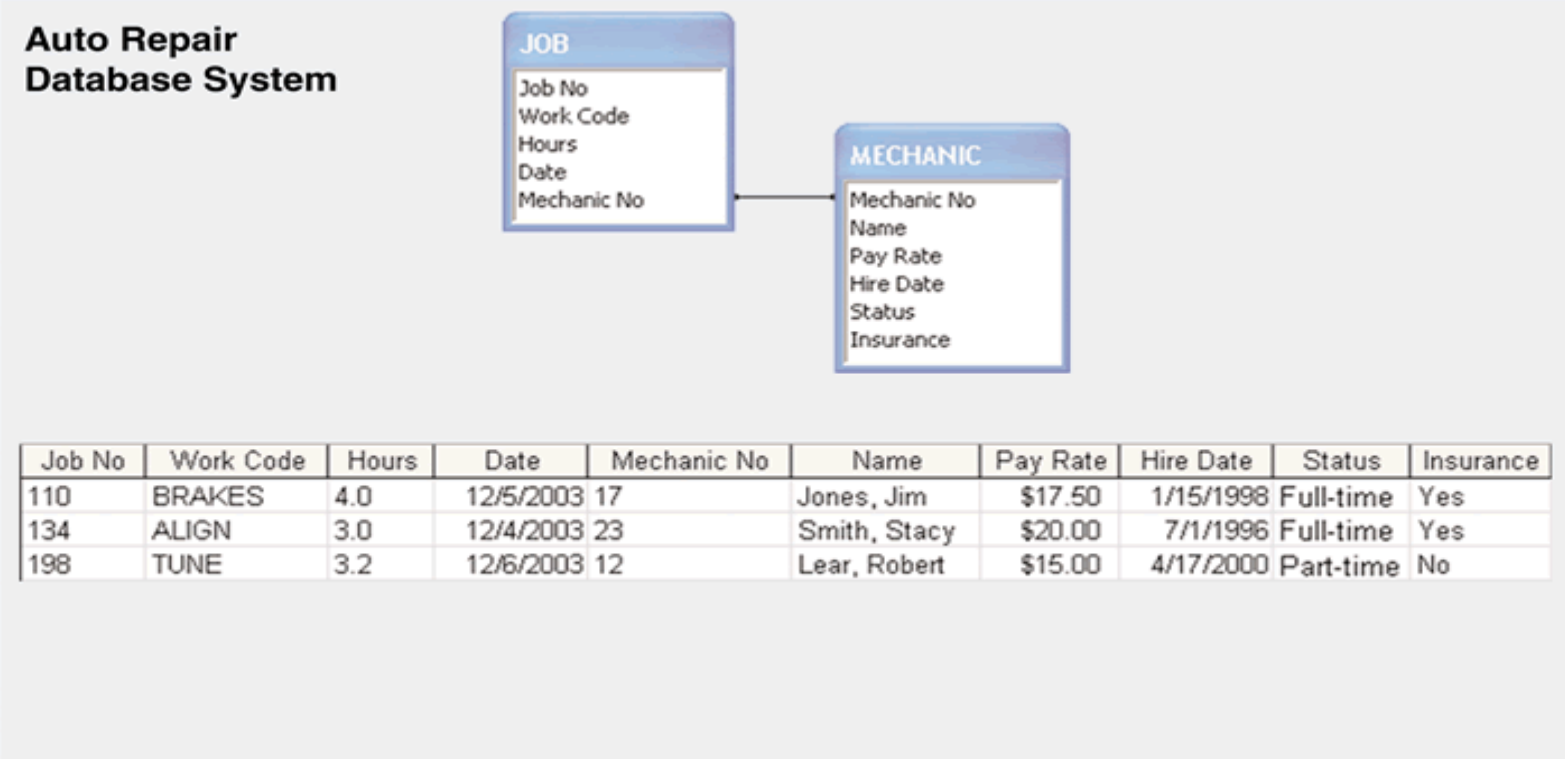
- Database: a collection of groupings of information that relate to each other in some way.
- A properly design database system offers a solution to the problems of file processing
- Provides an overall framework that avoids data redundancy and supports a real-time, dynamic environment
- In a Database environment several systems can be built and share a single database

Overview of Database Systems

A typical database environment might consist of a database serving five separate business systems.



Data Design Concepts



Job No	Work Code	Hours	Date	Mechanic No	Name	Pay Rate	Hire Date	Status	Insurance
110	BRAKES	4.0	12/5/2003	17	Jones, Jim	$17.50	1/15/1998	Full-time	Yes
134	ALIGN	3.0	12/4/2003	23	Smith, Stacy	$20.00	7/1/1996	Full-time	Yes
198	TUNE	3.2	12/6/2003	12	Lear, Robert	$15.00	4/17/2000	Part-time	No

Example of a database design for the auto repair shop that links two data tables and avoids duplication. Notice that the Mechanic No field provides a link between the two tables, and information can be accessed from either table.

Overview of Database Systems

- A database management system (DBMS) is a collection of tools, features, and interfaces that enables users to add, update, manage, access, and analyze the contents of a database
- The main advantage of a DBMS is that it offers timely, interactive, and flexible data access

Overview of Database Systems

- Advantages
  - Scalability: Scalability means that a system can be expanded, modified, or downsized easily to meet the rapidly changing needs of a business enterprise. Also known as extensibility.
  - Better support for client/server systems: In a client/server system, processing is distributed throughout an organization. Client/server systems require the power and flexibility of database design.
  - Economy of scale: The inherent efficiency of high-volume processing on larger computers. Database design allows better utilization of hardware. If a company maintains an enterprise-wide database, processing is less expensive using a powerful mainframe server instead of using several smaller computers

Overview of Database Systems

- Advantages
  - Flexible data sharing: Data can be shared across the enterprise, allowing more users to view the same information in different ways
  - Enterprise-wide application – database administrator (DBA): (DBA) typically manages a database management system (DBMS). The DBA assesses overall requirements and maintains the database for the benefit of the entire organization rather than a single department or user.
  - Stronger standards: effective database administration helps ensure that standards for data names, formats, and documentation are followed uniformly throughout the organization

Overview of Database Systems

- Advantages
  - Controlled redundancy: because the data is stored in a set of related tables, data items do not need to be duplicated in multiple locations
  - Better security: the DBA defines authorization procedures to ensure that only legitimate users can access the database and can allow different users to have different levels of access
  - Increased programmer productivity: programmers do not have to create the underlining file structure for the database, so they can concentrate on logical design
  - Data independence: systems that interact with a DBMS are relatively independent of how the

Database Tradeoffs

- Because DBMSs are powerful, they require more expensive hardware, software, and data networks capable of supporting a multi-user environment
- More complex than a file processing system
- Procedures for security, backup, and recovery are more complicated and critical